REMARKS/ARGUMENTS

In the Final Office Action dated December 7, 2005, Claims 1-6, 12-14, 23-26 and 28-29 stand rejected for anticipation by US Patent 6,789,252 granted to Bruke.

The Final Office Action explained the anticipation rejection of Claim 1 in paragraph 6 starting at the top of page 3. Specifically, Claim 1s' "creating a new object for use by a new instance of the application" was rejected over Burke's column 9 lines 47-50, column 5 lines 31-59, column 18 lines 65-67, column 19 lines 1-5, column 20 lines 14-32 and column 79 lines 1-5. The just-described lines from Burke's patent are reproduced:

> **column 9 lines 47-50**
>
> FIG. 7 graphically depicts an example of how the clone method produces a new object by copying an existing object.
>
> FIG. 8 graphically depicts an example of how the compose method melds together a base specification and a supplemental specification to produce a composite specification.
>
> **column 5 lines 30-59**
>
> SUMMARY
>
> To achieve the foregoing goals, and in accordance with the purposes of the invention as embodied and broadly described in this document, we have provided a method and system for creating and applying business objects that take the form of dynamically managed definitions. The method and system is based on an open and extensible object definition framework that manages definitions as specifications. The method and system provides the means for defining any business object needed in a business system regardless of its scope and type.
>
> In accordance with the invention, an open and extensible object definition framework is provided for managing business object definitions as specifications. The framework is based on the assumption that that the definition of anything and everything will evolve over time. Attributes of business objects that are not relevant when a system is first used may become relevant as the object finds new applications.
>
> All object definitions/specifications have the potential of interrelating with other definitions/specifications within and across company boundaries. This produces an urgent mandate for a definition system that can be configured for an industry or industry group to provide a common ontology a collaborative environment for applying and exchanging object definitions and specifications. Such a system must support end user creation of definitions/specifications through a consistent user interface for the viewing, understanding, exchanging, and processing said definitions/specifications.

> **column 18 lines 65-67 and column 19 lines 1-5**
> The invention can be used to define templates for the creation and management of Business Objects. As discussed above, these templates are known as "models". Models are business objects that take the form of revision-controlled specifications. Model definitions are composed of other business objects that also are dynamically created and/or composed by users of the invention.
>
> **column 20 lines 14-32**
> The Create function initializes new business object instances by instantiating them from a selected Model. The newly instantiated instance then has its definition completed in accordance with criterion that it has inherited from its parent Model.
> ...
> The clone method instantiates a new object from an existing object (specification) providing the newly instantiated object with a new identifier. FIG. 7 graphically depicts an example of how the clone method produces a new object by copying an existing object. The newly instantiated object is given an open pending revision. While the pending revision is open, the new object's definition can be modified. The collection and derivation rules that govern the new object's specification are the same as existed for the source object. Stated another way, the new object inherits the source object's model parentage.
>
> **column 79 lines 1-5**
> 204. The method according to claim 203 further comprising running the renew method to refresh in mass a plurality of existing instances of business objects instantiated from a given parent model template in response to a user request.

Note that all of the above-quoted text in Burke's patent refers to creating an instance of an object, which is commonly done by cloning, in the art of object oriented programming.

Hence, in Burke's patent, a typical object instance is an instance of a datatype (class), such as the Java class "Integer" e.g. Integer i, j; wherein i and j are instances of the class (type) Integer. Burke explicitly states that "[I]n a preferred embodiment, the invention is implemented in the Java programming language, relying substantially on its ***object-oriented programming*** techniques" (emphasis added; see Burke's column 30 lines 50-52).

Claim 1 is believed to be patentable over Burke's patent for numerous reasons, although only eight such reasons are discussed below.

A ***first*** distinction over Burke's patent is that an instance of an application as recited in Claim 1 is not the same as (and is therefore not disclosed or suggested by) an instance of an object in object oriented programming (OOP). For examples of application instances, see Applicants' originally-filed specification, at page 1 lines 23-26 which describes "multiple instances (e.g. Instance1 and Instance 2 in FIG. 1) of the database server". Also see the specification at page 2 at lines 21-23 which states that "a single instance of a database process (also called "Oracle Instance") is executing on each of the computers ..."

This meaning of the word "instance" in Claim 1 was clarified in the prior Amendment. Claim 1's instance is explicitly described as follows: "wherein each instance of the application comprises a plurality of processes." This limitation of Claim 1 was rejected in the Final Office Action, for anticipation by Burke's column 6 lines 10-30 and column 5, lines 30-59. Column 6 lines 10-30 are reproduced below, whereas column 5, lines 30-59 have already been reproduced in the current Response at page 2 hereof.

> **column 6 lines 10-30**
> The invention provides a shared vision across a company out into the marketplace for the exchange and collaboration of the definitions and specifications that are the basis for web based commerce. It provides a framework for an integrated solution covering the entire value chain to provide a total solution. An integrated solution applying to the cycles of creating new products and bringing them to market as well as to the interactions of buyers and sellers in the market. The invention provides the ability to exchange and compare and collaborate on specifications, on which the success of supply chain and web based market interactions will depend.
>
> The invention may be used to define any object that is to be processed by a computer. Objects can include Properties, Classifications, Knowledge, Business Objects, and Business Rules to name a few. Typical Business Objects include: Business and social entities; Locations including spaces, places, and channels; Activity including events and processes; Items including products and services; Business Records including orders and other forms of demand, inventory, jobs, deliverables, statements, transaction history et. al.

Nothing in the just-quoted text discloses or suggests an instance of an application, of the type recited in Claim 1. Burke's object instance which is created by OOP cloning fails to disclose or suggest an application instance which contains multiple processes as recited at the end of Claim 1. Hence, a multi-process application is nowhere disclosed or suggested in the entirety of Burke's patent. Moreover, Burke's patent fails to disclose or suggest multiple instances, of such a multi-process application.

As previously noted, the term "object" as recited in Claim 1 is used in its generalized sense. Specifically, Claim 1's object has nothing to do with object oriented programming (i.e. Claim 1 does not require nor does it preclude use of an object inheritance mechanism). This interpretation of the term "object" is apparent to a skilled artisan from the originally-filed specification, e.g. at page 6 lines 2-11.

A ***second*** distinction over Burke's patent is that Claim 1 does not explicitly recite creation of a new instance (i.e. a new set of processes) of an application. Instead, Claim 1 recites that a new object is created, from an existing object that is used by an existing instance (i.e. an existing set of processes) of the application. Nothing in the above-quoted text from Burke's patent discloses or suggests creating a new object outside of the context of a process. As noted above, Burke's patent at most discloses creation of object instances, by OOP cloning, which results in objects within the context of the process (which created the objects), because the cloned object inherits the source object's model parentage (see Burke's column 20 line 32). Hence, Burke fails to disclose or suggest creating a new object outside of process context, which is required if an existing application instance's existing object is used to create a new object for a new application instance, as recited in Claim 1. Specifically, Claim 1's creating is performed outside of the new application (which is to be started), a concept nowhere disclosed or suggested by Burke.

Claim 1 requires that after creation of the new object, this newly-created object is used (by the new application instance) to access a resource that is shared by multiple instances (i.e. multiple sets of processes) of the application. The multiple instances in Claim 1 include the existing instance whose existing object has been used to create the new object. Hence a ***third*** distinction over Burke's patent is that Burke fails to disclose or suggest a resource shared by multiple instances of the multi-process application. This limitation in Claim 1 was not addressed in the Final Office Action, which therefore fails to make a prima facie rejection (see top half of page 3 of the Final Office Action).

A ***fourth*** distinction over Burke's patent is that there is no disclosure or suggestion (in the above-quoted text from Burke's patent) for the shared resource to be accessed by the new application instance using the newly created object. This limitation in Claim 1 was also not addressed in the Final Office Action. This is an additional reason why the Final Office Action fails to make a prima facie rejection, and must be withdrawn.

In view of the above four arguments, Applicants respectfully submit that Claim 1's "creating" limitation (when interpreted as described in the wherein clauses thereof) is nowhere disclosed or suggested by the Burke patent. Although to overcome an anticipation rejection, it is sufficient to show that a single claimed limitation is not found in the cited reference, Applicants respectfully submit that Claim 1's remaining limitations are also not disclosed or suggested by Burke, as discussed next.

Specifically, the Final Office Action states that Claim 1's "setting up a connectivity between the new instance and the network" is disclosed in Burke's patent at column 25, lines 6-14 and column 26 lines 1-16 which are reproduced below:

> **column 25 lines 6-14**
> The Enterprise Explorer software component allows a user to all definitional content of an object from one user interface. The user can selectively view the object's revision, composition, attributes, value refinements, classifications, network connections, renderings, or reference information. The user can execute the following revision actions against the selected component: Open, Release, Cancel, Freeze and Unfreeze. The user can also execute the following functions against the selected component: Create, ...
>
> **column 26 lines 1-16**
> supports Web based interaction between suppliers and customers using business object definition components and system embodiments. Business object definition components are used to support the evolutionary stages of two party collaboration, including tandem collaboration, customer collaboration and co-development collaboration. Each of these stages of collaboration is described below.
>
> The invention's tandem collaboration functionality allows a supplier to take Internet enabled orders real-time, while a customer watches or verifies the building of the order and the product spec during customer's order entry. Historically, Customer Service Representatives (CSR's) have collaborated with customers via phone. Using the tandem collaboration functionality of the invention, a customer may simultaneously be in communication via phone and via a computer network, such as the Internet.

Nothing in the above-quoted text even remotely suggests the setting up of connectivity between an instance of a multi-process application, and a network. Instead, all of this text by Burke at most suggests use of pre-existing network connections to support interactions between suppliers and customers using business object definition components. As will be apparent to the skilled artisan, suppliers and customers are two specific entities, and therefore Burke merely teaches their interactions, and permits a user to view an object's properties. However, the claimed setting up of connectivity between the new application instance and a network is nowhere disclosed or suggested by Burke. Hence, this is a ***fifth*** distinction over Burke's patent.

The Final Office Action also states that Claim 1's "starting execution of a new instance of the application" is disclosed in Burke's patent at column 25 lines 38-50, column 56 lines 62-65, and column 4 lines 37-42, all of which are reproduced below:

> **column 25 lines 38-50**
> The invention allows use of ingrediential composition to create objects in a tree space of nodes representing knowledge. The usual generic name that is applied is "enterprise knowledgebase". The objects in this tree space can be traversed up and down. They can also be launched, resulting in execution or display of the launched node object. FIGS. 75A through 75C show an example of products at different levels in such a knowledge hierarchy. Some of the nodes in the example knowledge hierarchy are search objects that when launched (e.g., by double clicking with an input mouse) result in a display of a search template, such as the display shown in FIG. 76. At this point, the user may optionally change or further refine the search criteria ...
>
> **column 56 lines 62-65**
> 30. The method according to claim 25 further comprising attributing one or more ingrediential objects in an object specification template as executable logic expressions by the steps including: ...
>
> **column 4 lines 37-42**
> Another goal of the invention is to provide a system and method that can define, program, configure and execute a set of business objects that provide a common model of people, places, things and activities, the commerce interactions between them, the rules that govern those interactions and the business information models that result from these interactions..

Nothing in the above-quoted text even remotely suggests the start up of a multi-process application instance. Instead, Burke at most suggests that one or more objects may be launched, resulting in execution. As would be apparent to the skilled artisan, the startup of multiple processes of an application instance is not disclosed or suggested merely by the execution of an OOP object. This is a ***sixth*** distinction over Burke's patent.

Moreover, although several of the words in Claim 1 are found in Burke's patent, Applicants respectfully submit that Burke's words are used in a different context and with a different meaning than the words in Claim 1. As one example, Burke's instances are instantiations of OOP objects, whereas Claim 1's instances are instantiations of a multi-process application. As another example, Burke's applications are business applications (such as a product composition system, an order management system, and a customer ordering system), whereas Claim 1's application can be any application in a computer that comprises multiple processes. Therefore, this is a ***seventh*** argument for patentability of Claim 1 over Burke's patent.

Even if Burke's words were to be used in the same context as and with the same meaning as Claim 1, Burke's steps have different interrelationships among each other and result in a different combination than the method recited in Claim 1. As stated by the Court of Appeals for the Federal Circuit, "An anticipating reference must describe the patented subject matter with sufficient clarity and detail to establish that the subject matter existed and that its existence was recognized by persons of ordinary skill in the field of the invention." See ATD CORPORATION v/s LYDALL, INC. (Fed. Circ. 1998), citing In re Spada, 911 F.2d 705, 708, 15 USPQ2d 1655, 1657 (Fed. Cir. 1990) and Diversitech Corp. v. Century Steps, Inc., 850 F.2d 1566, 1567, 7 USPQ2d 1315, 1317 (Fed. Cir. 1988). Burke's patent simply fails to disclose or suggest the identical method recited in Claim 1. Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989); MPEP § 2131. This is therefore an ***eighth*** distinction over Burke's patent.

Since Burke fails to disclose or suggest Claim 1, for at least one or more of the eight reasons noted above, Applicants respectfully request the Examiner to withdraw the prior art rejection of Claim 1. Claims 2-14 and 26-28 depend from Claim 1 and are patentable for at least the same reasons as Claim 1.

The Examiner rejected the dependent claims, for several reasons that are hereby respectfully traversed, i.e. not accepted by the Applicants. Although the Examiner's remarks regarding these claims have been rendered moot, in view of the above-described patentability of these dependent claims (for the same reasons as Claim 1), Applicants will now discuss some of these remarks to explain the irrelevance of Burke's teachings.

Burke fails to disclose or suggest making a copy of an existing object, to rename the copy as per Claim 2. The Examiner-cited text at column 20 lines 21-26 and column 54 at line 9 in Burke's patent fails to anticipate Claim 2 for at least two reasons. First, the clone being made by Burke appears to be an instance of an object which is accessed using its "new identifier" (column 20 line 23). Burke's "new identifier" appears to be a reference to a memory location, which memory reference is typically local to the process context, and is therefore not available for use by another application instance as claimed. Second, Burke's use of a "new identifier" teaches away from changing the name of an object. While Burke does disclose that properties of an object may be edited, Burke fails to disclose or suggest that the object itself is to be renamed. Therefore, there is no suggestion to use the name of a multi-process application while renaming the copied object, as recited in Claim 2. Claim 2 is therefore patentable for at least this additional reason.

Applicants also traverse the Examiner's statement in the middle of page 4 of the Office Action that Burke discloses Claim 14 at column 25 lines 6-25, reproduced below:

> **column 25 lines 6-25**
> The Enterprise Explorer software component allows a user to all definitional content of an object from one user interface. The user can selectively view the object's revision, composition, attributes, value refinements, classifications, network connections, renderings, or reference information. The user can execute the following revision actions against the selected component: Open, Release, Cancel, Freeze and Unfreeze. The user can also execute the following functions against the selected component: Create, Clone, Compose, Compare, Applicability Determination, Capability Assessment, Derive, Renew and Delete. A new component can be added (save) and the description can be changed (save) from this user interface.
>
> The Network Editor serves as a tool for graphically creating and maintaining network definition. The Network Editor uses drag and drop tactics for visually organizing the network. The most common use of the Network Editor is in defining business processes and product routings. The Network Editor stores the result as ingrediential objects of the business object definition.

Nothing in the above-quoted text even remotely suggests a map file that is shared across all computers, as recited in Claim 14. Moreover, nothing in this text discloses or suggests the addition of an entry for the new instance in such a map file, that identifies which

SILICON VALLEY
PATENT GROUP LLP
[illegible] Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

instances are running on which computers (e.g. which application instance is running on which computer). Claim 14 is therefore patentable for at least this additional reason.

Claim 29 was rejected over the teachings of Burke's column 33 lines 50-60 and column 13 lines 55-60 which are reproduced below.

> **column 33 lines 50-60**
> Windows NT Application Server 4.0 SP4; and Java 1.1.8. Object-to-relational mapping is provided by middleware, such as Toplinks, which maps between the object definition and relational view. The system of FIGS. 31 and 32 is a message-based system where a single client can communicate to any number of server side components. The switchboard acts as a classic switchboard operator. When a message is sent to it, it routes the message to the appropriate component, and then sends the response back to the initiating client.

> **column 13 lines 55-60**
> the transport medium between the client ORB and the server ORB, a mapping is required from the GIOP to the TCP/IP level (which is the basis for Internet traffic). This mapping is known as the Internet Inter-Orb Protocol (IIOP).

Burke's column 33 quoted above teaches object-to-relational mapping, i.e. transformation between OOP and relational models of computer programming. Burke's column 13 quoted above teaches protocol mapping, i.e. conversion of a message that conforms to one protocol into another protocol. Applicants respectfully submit that both of the above-quoted teachings by Burke fail to disclose or suggest a map file that identifies a mapping between application instances and computers in a cluster (as noted above, which application instance is running on which computer in the cluster). Claim 29 is therefore patentable for at least this additional reason.

Note that the above-described rejection of Claim 29 is symptomatic of a problem with all claim rejections in the Final Office Action. Specifically, the only thing common among Columns 33 and 13 of Burke and Claim 29 is that all of them use the word "map". However, this word "map" is used in three different meanings, in column 33, column 13, and Claim 29. Applicants respectfully submit that the mere presence of the same words in Burke's patent is insufficient to support the anticipation rejection of Applicants' claims. The only thing that appears to be common between Applicants' claims and Burke's patent appears to be the presence of some words, which words are used in different contexts, with different meanings, and are interrelated to one another in different ways.

Apparatus Claim 23 was rejected in the Final Office Action for the same reasons as Claim 1, in the top half of page 3 of the Office Action. The rejection of Claim 1 has been addressed above, and hence Applicants have also implicitly addressed the rejection of Claim 23, i.e. Claim 23 is believed to be patentable over Burke's patent for numerous reasons, similar to reasons discussed above in reference to Claim 1. Specifically, Claim 23 distinguishes over Burke's patent for at least arguments similar to the first, second, third, fourth, fifth, sixth, and eighth arguments for the patentability of Claim 1.

Applicants respectfully submit that the obviousness rejections in the Final Office Action, of dependent Claims 7, 10 and 27, based on combination of Burke's teachings with Feurman's teachings or with Snyder's teachings are also without merit, for at least the above-described reasons for the patentability of independent Claim 1.

Also, Claim 7 requires displaying a list and receiving a selection from the list which is nowhere disclosed or suggested by Burke's patent, as admitted at the top of page 5 of the Final Office Action. In fact Burke's patent teaches away from using Claim 7's list, by disclosing a Network Editor for graphically creating and maintaining a network definition (see Burke's column 25 at lines 19-25). The Final Office Action stated that a skilled artisan would be motivated to modify Burke's patent so the user can view Feuerman's list of elements at a single time on a computer screen in a user-friendly environment. The Final Office Action has not explained why Burke's Network Editor does not already provide a single screen and user-friendly environment as shown in FIG. 40A and 40B in Burke's patent. So, what is the Examiner's reason to replace Burke's graphical interface with a list?

Furthermore, adding Feuerman's list to Burke's patent will require that Burke's patent be modified to support the "subscriber" feature of Feuerman's patent. Such a modification would require a re-design of Burke's system, thereby teaching away from the Examiner-proposed combination.

Moreover, as per the proposed motivation, Feuerman's list can be added to any prior art reference whatsoever if it contains multiple "elements". This proposed motivation is so generic and overbroad that it is insufficient to support an obviousness rejection of Claim 7. The Court of Appeals for the Federal Circuit has stated that a person of ordinary skill in the art must not only have had some motivation to combine the prior art teachings, but some motivation to combine the prior art teachings in the particular manner claimed.

See, e.g., In re Kotzab, 217 F.3d 1365, 1371 (Fed. Cir. 2000) ("Particular findings must be made as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed."); In re Rouffet, 149 F.3d 1350, 1357 (Fed. Cir. 1998) ("In other words, the examiner must show reasons that the skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention, would select the elements from the cited prior art references for combination in the manner claimed.").

Claim 7 (and Claim 10 that depends thereon) is therefore patentable for at least these additional reasons.

Claim 10 was rejected over the teachings of Feuerman's patent in column 4, lines 57-62 which are reproduced below:

> The user can select one of the names from the drop-down list, and the data delivery program will direct the data file from the scanner to the IP address and port number of the workstation that is associated with the human-readable name. The data delivery program can also be installed by a human administrator.

Nothing in the above-quoted text discloses or suggests any checking, prior to installation of the data delivery program. The Final Office Action states, without basis, that the skilled artisan would be motivated to check for presence of system resources that are vital to the system. This statement is hereby respectfully traversed for being not supported in the prior art. Moreover, an explicit limitation in Claim 10 "if said second computer does not have said software" is nowhere disclosed or suggested by Feuerman. Claim 10 is therefore patentable for at least these additional reasons.

Claim 27 was rejected over the teachings of Snyder's patent in column 14, lines 13-24 which are reproduced below:

> Creation functions are static member functions that are public. These can be called from other code in the server, such as a registration hook (described above) or a factory object method. Creation functions are not called through the interface class, but rather are called directly by name. In one embodiment, the Creation functions include a Creator function to create an instance of a servant class and its persistent data which the creator function stores in the refdata of the object. The creator function also signals the OA to create the object and calls any initialization hooks that have been defined by the programmer (see above),

Note that Snyder teaches storing the persistent data in the "refdata" of his OOP object. Therefore, Snyder fails to disclose or suggest that the persistent data is for a multi-process instance of an application, i.e. Snyder fails to cure Burke's defect as per the first argument above. Moreover, Snyder fails to disclose or suggest that his persistent data is to be stored in a resource that is shared by multiple instances of an application.

The Final Office Action stated that a skilled artisan would be motivated to modify Burke's patent to implement a permanent resource configuration to operate the system properly. The Final Office Action has not explained why a Configuration System which includes all attributes cannot be used as described by Burke at column 47, line 16. Furthermore, adding Snyder's Creation functions to Burke's patent will cause confusion because Burke already discloses a "Create function" (at column 20, lines 14-18). Therefore, Snyder's Creation functions if replacing or adding to Burke's system would require a re-design, thereby teaching away from the Examiner-proposed combination.

Moreover, as per the proposed motivation, Snyder's Creation functions can be added to any prior art reference "to operate the system properly." This proposed motivation is so generic and overbroad that it is insufficient to support an obviousness rejection of Claim 27. As noted above, particular findings must be made for some motivation to combine in the particular manner claimed, which is not present in the Final Office Action. Also, the Examiner has not explained why Burke's system doesn't operate "properly", i.e. what is improper in Burke's system to require use of Snyder's Creation functions.

In view of the above, Applicants believe all Claims 1-14 and 23-29 are patentable over the cited references. Therefore, Applicants respectfully request allowance of all pending claims. Should the Examiner have any questions concerning this response, the Examiner is invited to call the undersigned at (408) 982-8203.

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted to the U.S. Patent and Trademark Office to the fax number 571-273-8300 on March 7, 2006 .

S. Omkar — March 7, 2006
Attorney for Applicant(s) — Date of Signature

Respectfully submitted,

S. Omkar

Omkar K. Suryadevara
Attorney for Applicant(s)
Reg. No. 36,320